

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ПРИДНІПРОВСЬКА ДЕРЖАВНА АКАДЕМІЯ БУДІВНИЦТВА ТА АРХІТЕКТУРИ»**

Кафедра прикладної математики та інформаційних технологій


«ЗАТВЕРДЖУЮ»
Проректор з науково-педагогічної
та навчальної роботи
Р. Б. Цапирник
« 15 » *жовтня* 2019 року

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Архітектура та проектування програмного забезпечення

| | |
|------------------------------|-----------------------------|
| спеціальність | 122 «Комп'ютерні науки» |
| освітньо-професійна програма | «Комп'ютерні науки» |
| освітній ступінь | бакалавр |
| форма навчання | денна |
| розробник | Семенець Сергій Миколайович |

1. АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Дисципліна «Архітектура та проектування програмного забезпечення» є нормативною компонентою циклу професійної підготовки бакалаврів за спеціальністю 122 «Комп'ютерні науки». Предметом вивчення дисципліни є архітектура та методи проектування програмного забезпечення (ПЗ). Вивчення даної дисципліни дає знання, що використовуються на етапі проектування ПЗ при розробці програмного дизайну.

Згідно з навчальною програмою дисципліни розглядаються наступні основні питання: класифікація ПЗ; архітектура ПЗ; архітектурні компоненти; архітектурні шаблони та моделі; етапи та моделі життєвого циклу програмного продукту; управління процесом проектування ПЗ; класифікація вимог до ПЗ; атрибути якості; аналіз вимог та документування; стратегії і методи проектування ПЗ; об'єктно-орієнтоване, функціонально-орієнтоване та компонентне проектування ПЗ; архітектурні фреймворки; case- технології проектування ПЗ; принципи об'єктно-орієнтованого проектування; аналіз та оцінка якості програмного дизайну; уніфікована мова моделювання UML; діаграми UML; патерни проектування ПЗ; підвищення проектної надійності інформаційних систем.

2. ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| | Години | Кредити | Семестр |
|---|--------|---------|---------|
| | | | V |
| Всього годин за навчальним планом, з них: | 135 | 4,5 | 135 |
| Аудиторні заняття, у т.ч: | 46 | | 46 |
| лекції | 30 | | 30 |
| лабораторні роботи | 16 | | 16 |
| практичні заняття | | | |
| Самостійна робота, у т.ч: | 89 | | 89 |
| підготовка до аудиторних занять | 17 | | 17 |
| підготовка до контрольних заходів | 2 | | 2 |
| виконання курсового проекту | 30 | | 30 |
| опрацювання розділів програми, які не викладаються на лекціях | 10 | | 10 |
| підготовка до екзамену | 30 | | 30 |
| Форма підсумкового контролю | | | Екзамен |

3. СТИСЛИЙ ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета дисципліни - теоретична та практична підготовка фахівців з питань архітектури і проектування програмного забезпечення інформаційних систем.

Завдання дисципліни - опанування студентами сучасними методами проектування програмного забезпечення; отримання знань про архітектуру, методи аналізу та розробки програмного дизайну; отримання практичних навичок з проектування програмних продуктів.

Пререквізити дисципліни. «Інформатика», «Алгоритмізація і програмування»

Постреквізити дисципліни. «Об'єктно-орієнтоване програмування», «Проектування інформаційних систем».

Компетентності.

Інтегральна компетентність:

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук, або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності:

ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу

ЗК-2. Здатність застосовувати знання у практичних ситуаціях.

ЗК-3. Знання та розуміння предметної області та розуміння професійної діяльності

ЗК-6. Здатність вчитися і оволодівати сучасними знаннями.

ЗК-7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК-12. Здатність приймати обґрунтовані рішення.

Спеціальні (фахові, предметні) компетентності:

СК-8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: структурного, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.

Заплановані результати навчання:

РН-1. Здобувати систематичні знання в галузі комп'ютерних наук, аналізувати проблеми з точки зору сучасних наукових парадигм, осмислювати і робити обґрунтовані висновки з наукової і навчальної літератури та результатів експериментів.

РН-20. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та

алгоритмів розв'язання задач в галузі комп'ютерних наук, створювати надійне та ефективне програмне забезпечення.

РН-27. Використовувати технології проектування складних систем, вибирати CASE- засоби; формулювати техніко-економічні вимоги, розробляти інформаційні та програмні системи з використанням шаблонів та засобів автоматизованого проектування.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- архітектурні шаблони та моделі програмних систем;
- моделі життєвого циклу програмного продукту;
- вимоги до програмного забезпечення;
- стратегії і методи проектування програмного забезпечення;
- архітектурні фреймворки та CASE- технології проектування;
- принципи об'єктно-орієнтоване проектування;
- уніфіковану мову моделювання UML;
- критерії оцінювання якості дизайну програмного забезпечення;
- патерни проектування програмного забезпечення;
- методи підвищення проектної надійності інформаційних систем.

вміти:

- орієнтуватися в типах архітектури сучасного програмного забезпечення;
- оцінювати якість програмного дизайну;
- аналізувати вимоги та розробляти відповідний дизайн програмного забезпечення;
- розробляти проектну документацію для опису та підтримки програмного забезпечення;
- управляти процесом проектування програм;
- проектувати програмні продукти згідно з наданим технічним завданням.

Методи навчання: словесні методи (лекція); наочні методи (ілюстрація, демонстрація); практичні методи (вправа, практична робота).

Форми навчання: фронтальні, групові, аудиторні, позааудиторні.

4. СТРУКТУРА (ТЕМАТИЧНИЙ ПЛАН) ДИСЦИПЛІНИ

| Назва змістових модулів і тем | Кількість годин, у тому числі | | | | |
|--|-------------------------------|-----------|---|-----------|-----------|
| | усього | л | п | лаб | с/р |
| Змістовий модуль 1. Архітектура та проектування ПЗ | | | | | |
| Класифікація програмного забезпечення. | 4 | 2 | | | 2 |
| Архітектура ПЗ. | 7 | 4 | | | 3 |
| Етапи розробки програмного продукту. | 5 | 2 | | | 3 |
| Вимоги до ПЗ. | 5 | 2 | | | 3 |
| Стратегії і методи проектування ПЗ. | 11 | 2 | | 6 | 3 |
| Принципи об'єктно-орієнтоване проектування. | 8 | 4 | | | 4 |
| Уніфікована мова моделювання UML. | 18 | 6 | | 6 | 6 |
| Патерни проектування. | 7 | 4 | | | 3 |
| Підвищення проектної надійності ІС . | 10 | 4 | | 4 | 2 |
| Разом за змістовим модулем 1 | 75 | 30 | | 16 | 29 |
| Змістовий модуль 2. Курсовий проект | | | | | |
| Аналіз вимог та декомпозиція системи. | 4 | | | | 4 |
| Обрання та освоєння патерну проектування з урахуванням вимог до системи. | 4 | | | | 4 |
| Побудова діаграми варіантів використання UML. | 2 | | | | 2 |
| Побудова діаграми класів UML. | 4 | | | | 4 |

| | | | | | |
|---|-----|----|--|----|----|
| Побудова діаграми послідовності UML. | 4 | | | | 4 |
| Аналіз та оцінка якості програмного дизайну. | 2 | | | | 2 |
| Розрахунок показників надійності системи. | 2 | | | | 2 |
| Розробка моделі оптимального резервування системи з урахуванням надійності. | 2 | | | | 2 |
| Розробка оптимальної структурної схеми надійності системи. | 2 | | | | 2 |
| Оформлення пояснювальної записки. | 4 | | | | 4 |
| Разом за змістовим модулем 2 | 30 | | | | 30 |
| Підготовка до екзамену | 30 | | | | 30 |
| Усього годин | 135 | 30 | | 16 | 89 |

5. ЛЕКЦІЙНИЙ КУРС

| № занять | Тема занять | Кількість годин |
|----------|---|-----------------|
| 1 | Класифікація програмного забезпечення. Предмет та задачі дисципліни. Програмне забезпечення (ПЗ) комп'ютера. Системне ПЗ. Операційні системи та драйвери. Прикладне ПЗ. Текстові і графічні редактори, табличні процесори, системи управління базами даних, мультимедійні програми та САПР. Службове ПЗ. Утиліти та мережеві протоколи. Інструментальне ПЗ. Редактори, компілятори та бібліотеки мов програмування. | 2 |
| 2,3 | Архітектура ПЗ. Визначення архітектури програмного забезпечення. Архітектурні компоненти. Архітектурні шаблони та моделі. Структурний та поведінковий описи архітектури. Монолітна архітектура. Плагінна та мікро-сервісна архітектури. Багатошарова та багаторівнева архітектури. Рівні представлення даних, прикладної логіки, управління та доступу до ресурсів, джерела даних. Архітектура «канали та фільтри». Об'єктно-орієнтована архітектура. Архітектурні моделі «Клієнт-Сервер» та MVC. Програми зі змішаною архітектурою. | 4 |
| 4 | Етапи розробки програмного продукту. Життєвий цикл програмного продукту (ПП). Основні етапи життєвого циклу ПП. Планування. Розробка технічного завдання (формування вимог до системи). Аналіз вимог та проектування. Етап конструювання. Інтеграція і тестування. Супровід і експлуатація. Моделі життєвого циклу ПП. Каскадна, спіральна та інкрементна моделі. Стороння розробка ПП. Заміна версії та управління конфігурацією системи. | 2 |
| 5 | Вимоги до ПЗ. Класифікація вимог до програмного забезпечення. Бізнес вимоги та бізнес правила. Вимоги користувача. Функціональні і не функціональні вимоги. Ефективність, надійність та безпека системи. Атрибути якості. Вимоги до даних. Вимоги до інтерфейсів. Аналіз вимог та документування. | 2 |
| 6 | Стратегії і методи проектування ПЗ. Загальні стратегії проектування ПЗ. Функціонально-орієнтоване (структурне) проектування ПЗ. Проектування "зверху-вниз" і "знизу-вгору". Компонентне проектування ПЗ. Об'єктно-орієнтоване проектування ПЗ. Архітектурні фреймворки. CASE - технології проектування. Нотації проектування. | 2 |
| 7,8 | Принципи об'єктно-орієнтованого проектування. Принципи SOLID як базові принципи об'єктно-орієнтованого проектування ПЗ. Принцип єдиної відповідальності. Принцип відкритості / закритості. Принцип підстановки. Принцип поділу інтерфейсу. Принцип інверсії залежностей. Інкапсуляція, абстракція, поліморфізм, спадкування та принципи SOLID. Критерії якісної | 4 |

| | | |
|-------|--|----|
| | та неякісної архітектури. Розширюваність, гнучкість, масштабованість, тестованість та ремонтпридатність програми. Можливість перевикористання модулів. Жорсткість, крихкість, нерухомість. Аналіз та оцінка якості програмного дизайну. | |
| 9-11 | Уніфікована мова моделювання UML. Мова UML як засіб моделювання ПЗ. Основні будівельні блоки UML. Структурні та поведінкові сутності. Сутності, що групують та сутності, що анотують. Діаграми UML. Діаграма класів як засіб моделювання статичної структури програмної системи в термінах класів та відношень між ними. Графічне зображення класів. Атрибути та операції (методи) класу. Абстрактні класи. Інтерфейси. Умовне графічне зображення відносин залежності, асоціації, композиції, агрегації, узагальнення та реалізації. Приклади складання діаграм класів. Діаграми варіантів використання, послідовності, станів та діяльності. Інші діаграми. | 6 |
| 12,13 | Патерни проектування. Призначення та класифікація патернів проектування. Породжуючи, структурні та поведінкові дизайн - патерни. Огляд дизайн - патернів Builder, Facade, Adapter, Decorator, Observer, Mediator. | 4 |
| 14,15 | Підвищення проектної надійності інформаційних систем. Визначення надійності ІС. Розрахунок показників надійності систем з логічно послідовним, логічно паралельним та змішаним з'єднанням елементів. Метод структурного резервування для підвищення надійності складних систем. Оцінка надійності резервованої системи. | 4 |
| | Усього годин | 30 |

6. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Практичні заняття навчальним планом не передбачені.

7. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

| № занять | Тема занять | Кількість годин |
|----------|---|-----------------|
| 1-6 | Розробка UML-діаграм. | 6 |
| 7-11 | Розробка програмного дизайну. | 6 |
| 12-15 | Проектування ІС з урахуванням надійності. | 4 |
| | Усього годин | 16 |

8. САМОСТІЙНА РОБОТА

| № занять | Вид роботи / Назва теми | Кількість годин |
|----------|---|------------------------|
| 1 | Підготовка до аудиторних занять. | 17 |
| 2 | Підготовка до контрольних заходів. | 2 |
| 3 | Опрацювання розділів програми, що не викладаються на лекціях: - Мова UML як засіб моделювання програмного забезпечення. - Архітектура веб- додатків. - 4+1 View Model of Software Architecture. - Шаблони проектування. | 10 2 2 3 3 |
| 4 | Виконання курсового проекту: - Аналіз вимог та декомпозиція системи. - Обрання та освоєння патерну проектування з урахуванням вимог до системи. - Побудова діаграми варіантів використання UML. - Побудова діаграми класів UML. | 30 4 4 2 4 |

| | | |
|---|---|----|
| | - Побудова діаграми послідовності UML. | 4 |
| | - Аналіз та оцінка якості програмного дизайну. | 2 |
| | - Розрахунок показників надійності системи. | 2 |
| | - Розробка моделі оптимального резервування системи з урахуванням надійності. | 2 |
| | - Розробка оптимальної структурної схеми надійності системи. | 2 |
| | - Оформлення пояснювальної записки. | 4 |
| 5 | Підготовка до екзамену. | 30 |

9. МЕТОДИ КОНТРОЛЮ

Основними методами контролю знань студентів є усний, письмовий і графічний контроль, а також методи самоконтролю та самооцінки.

10. ПОРЯДОК ТА КРИТЕРІЇ ОЦІНЮВАННЯ

Структура оцінювання видів навчальної роботи студента у кожному змістовому модулі

Змістовий модуль 1. Архітектура та проектування програмного забезпечення

| № п/п | Вид навчальної роботи студента | Максимальна кількість балів |
|--------------|--|---------------------------------------|
| 1 | Виконання практичних робіт: | 90 (30 балів × 3 практичні роботи) |
| | 1. Розробка UML-діаграм. | |
| | 2. Розробка програмного дизайну. | |
| | 3. Проектування ІС з урахуванням надійності. | |
| 2 | Контрольна робота | 10 (5 балів × 2 питання) |
| Разом | | 100 |

Змістовий модуль 2. Курсовий проект

| № п/п | Вид навчальної роботи студента | Максимальна кількість балів |
|--------------|---|-----------------------------|
| 1 | Виконання курсового проекту | |
| | 1-й проміжний контроль: | |
| | Аналіз вимог та декомпозиція системи. | 6 |
| | Обрання та освоєння патерну проектування з урахуванням вимог до системи. | 6 |
| | Побудова діаграми варіантів використання UML: | 6 |
| | Побудова діаграми класів UML: | 6 |
| | Побудова діаграми послідовності UML. | 6 |
| | Усього | 30 |
| | 2-й проміжний контроль: | |
| | Аналіз та оцінка якості програмного дизайну. | 6 |
| | Розрахунок показників надійності системи. | 6 |
| | Розробка моделі оптимального резервування системи з урахуванням надійності. | 6 |
| | Розробка оптимальної структурної схеми надійності системи. | 6 |
| | Оформлення пояснювальної записки. | 6 |
| | Усього | 30 |
| 2 | Захист курсового проекту | 40 |
| Разом | | 100 |

Критерії оцінювання лабораторних робіт

Максимальна кількість балів за виконання однієї лабораторної роботи – 30. Загальна кількість лабораторних робіт – 3.

Кількість балів «30» – ставиться, якщо студент у відведений час повністю виконав обсяг робіт відповідно до передбаченого варіанта. Розв'язання задач виконано логічно послідовно, отримано правильні результати. Робота оформлена охайно.

Кількість балів «20–29» – ставиться, якщо студент у відведений час повністю виконав обсяг робіт відповідно до передбаченого варіанта. Розв'язання задач виконано логічно послідовно, отримано в цілому правильні результати, однак мають місце несуттєві помилки, робота оформлена не досить охайно.

Кількість балів «10–19» – ставиться, якщо студент у відведений час неповністю виконав обсяг робіт відповідно до передбаченого варіанта, не всі отримані результати є правильними, робота оформлена неохайно.

Кількість балів «0–9» – ставиться, якщо студент у відведений час не виконав обсяг робіт відповідно до передбаченого варіанта, при розв'язанні задач мають місце суттєві помилки.

Критерії оцінювання контрольної роботи

Контрольна робота складається з 2 запитань. Максимальна кількість балів за відповідь на 1 запитання – 5.

Кількість балів «5» – ставиться студенту за повну, змістовну, логічно послідовну, правильну відповідь у письмово-графічній формі на питання контрольної роботи.

Кількість балів «4» – ставиться студенту за логічно послідовну, загалом правильну відповідь в письмово-графічній формі на питання контрольної роботи. Але окремі пункти відповідей не повністю розкривають суть питання і мають незначні помилки.

Кількість балів «2–3» – ставиться студенту за відповідь в письмово-графічній формі на питання контрольної роботи, в якій не повністю розкривається суть поставлених питань. У розв'язанні задач наявні суттєві помилки, що свідчать про недостатнє засвоєння студентом теоретичного та практичного матеріалу. Представлена відповідь має фрагментарний характер, слабо пов'язана з суттю поставленого питання, оформлена недбало і не дає повного уявлення про вірність кінцевих результатів.

Кількість балів «0–1» – ставиться студенту за відсутність конкретної відповіді в письмово-графічній формі на питання контрольної роботи. Відповідь носить поверхневий безсистемний характер, відсутня теоретична база у висвітленні поставленого питання, наявні грубі помилки, що свідчать про відсутність у студента мінімуму знань з дисципліни.

Критерії оцінювання курсового проекту

Максимальна оцінка за виконання і захист курсового проекту – 100 балів, у т.ч. виконання курсового проекту – 60 балів; захист курсового проекту – 40 балів.

Виконання курсового проекту.

Максимальна кількість балів за виконання курсового проекту – 60 балів. Кількість задач, що містить завдання на курсовий проект – 10. Максимальна оцінка за розв'язання однієї задачі – 6 балів.

Кількість балів «6» – ставиться, якщо задача розв'язана правильно і в повному обсязі.

Кількість балів «3–5» – ставиться, якщо задача розв'язана в повному обсязі. Отримано в цілому правильні результати, однак мають місце несуттєві помилки.

Кількість балів «0–2» – ставиться, якщо задача розв'язана неправильно або в неповному обсязі, мають місце суттєві помилки.

Захист курсового проекту.

Максимальна кількість балів при захисті курсового проекту – 40 балів.

Кількість балів «40» – ставиться, якщо під час захисту студент демонструє системний характер знань сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються чітко і логічно-послідовно. На поставлені викладачем питання надаються повні, глибокі, обґрунтовані відповіді.

Кількість балів «30-39» – ставиться, якщо під час захисту студент показує досить високий рівень знань сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються чітко і логічно-послідовно. На поставлені викладачем питання надаються в цілому правильні, обґрунтовані відповіді, однак мають місце несуттєві помилки або неточності.

Кількість балів «20-29» – ставиться, якщо під час захисту студент не показує глибоке розуміння усіх сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються достатньо чітко і логічно-послідовно, але на поставлені викладачем питання надаються неповні або необґрунтовані відповіді.

Кількість балів «10-19» – ставиться, якщо під час захисту студент показує поверхневе знання сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються нечітко і невпевнено. На поставлені викладачем питання надаються помилкові або необґрунтовані відповіді.

Кількість балів «0-9» – ставиться, якщо під час захисту студент не демонструє необхідне розуміння сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються безсистемно, відсутня логічна послідовність викладання. Студент неспроможний надати правильні відповіді на поставлені викладачем питання.

Критерії оцінювання знань студентів на екзамені

Максимальна кількість балів на екзамені – 100 балів.

В екзаменаційному білеті 4 питання.

Максимальна кількість балів за відповідь на кожне питання – 25.

25 балів – вичерпна відповідь на запитання.

16–24 балів – ставиться за змістовну, логічно послідовну, в цілому правильну відповідь в письмовій формі на питання екзаменаційного білета. Відповідь охайно оформлено, але окремі підпункти питання розкриті не в повному обсязі, мають місце незначні помилки.

10–15 балів – ставиться за відповідь в письмовій формі на питання екзаменаційного білета, якщо студент надав поверхову відповідь на питання екзаменаційного білета. Допущені суттєві помилки, відсутня логічна послідовність відповіді.

0–9 балів – ставиться студенту при відсутності конкретної відповіді в письмово-графічній формі на екзаменаційне питання. Відповідь носить безсистемний характер, відсутня теоретична база у висвітленні поставленого питання, наявні грубі помилки, що свідчить про відсутність у студента відповідних теоретичних знань.

Підсумкова оцінка з дисципліни визначається як середньоарифметичне між оцінками змістового модуля 1 та екзамену.

Порядок зарахування пропущених занять. Пропущені лекції та лабораторні заняття, незалежно від причини пропуску, студент відпрацьовує викладачеві (презентація) згідно з графіком консультацій.

11. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Руководство Microsoft по проектированию архитектуры приложений. 2-е издание. - Корпорация Microsoft, 2012. - 529с.

2. Фаулер М. Архитектура корпоративных программных приложений. - М: Вильямс, 2016. - 541с.
3. Басс Л. Архитектура программного обеспечения на практике / Л. Басс, П. Клементс, Р. Кацман. - СПб: Питер, 2012. - 574с.
4. Гагарина Л. Г. Введение в архитектуру программного обеспечения. Учебное пособие. - М.: Инфра-М, 2016. - 320 с.
5. Панюкова Т. А. Проектирование программных средств. - М.: Гостехиздат, 2012. - 364 с.
6. Sommerwill I. Инженерия программного обеспечения / И. Sommerwill - М: Вильямс, 2012. - 467с.
7. Назаров С.В. Архитектура и проектирование программных систем. - М.: Инфра-М, 2013. - 351 с.
8. Гамма Э., Хелм Р. Приемы объектно-ориентированного программирования. Паттерны проектирования. СПб: Питер, 2014. - 366с.

Допоміжна

1. Коуд П., Норт Д., Мейфилд М. Объектные модели. Стратегии, шаблоны и приложения. - М: Лори, 2015. - 230с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. - М: Бином, 2010, 560с.
3. Якобсон А. Унифицированный процесс разработки программного обеспечения. -
4. Лавріщева К.М. Програмна інженерія. Підручник. - К: ВНУ, 2011. - 435с.
5. Мацяшек Л. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML.- М: Вильямс, 2014. - 432с.
6. Gamma E. Design Patterns. СПб.: Питер, 2014. -511 с.

12. INTERNET-РЕСУРСИ

1. Компонентный подход в программировании [Электронный ресурс]. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=2778
2. Архитектурное проектирование программного обеспечения [Электронный ресурс]. – Режим доступа: <http://www.lib.tpu.ru/fulltext2/m/2011/m416.pdf>
3. Обзор паттернов проектирования [Электронный ресурс]. – Режим доступа: <http://www.lib.tpu.ru/fulltext2/m/2011/m310.pdf>

Розробник

(С. М. Семенець)

Гарант освітньої програми

(Н.М. Єршова)

Силабус затверджено на засіданні кафедри
прикладної математики та інформаційних технологій

Протокол від «12» жовтня 2019 року № 3